

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-298097

(43)公開日 平成5年(1993)11月12日

(51)Int.Cl.⁵

G 0 6 F 9/38

識別記号 庁内整理番号

3 8 0 A 9193-5B

F I

技術表示箇所

3 1 0 E 9193-5B

// G 0 6 F 11/14

3 1 0 N 7313-5B

審査請求 未請求 請求項の数1(全6頁)

(21)出願番号

特願平4-99694

(22)出願日

平成4年(1992)4月20日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 平井 規郎

鎌倉市大船五丁目1番1号 三菱電機株式
会社情報電子研究所内

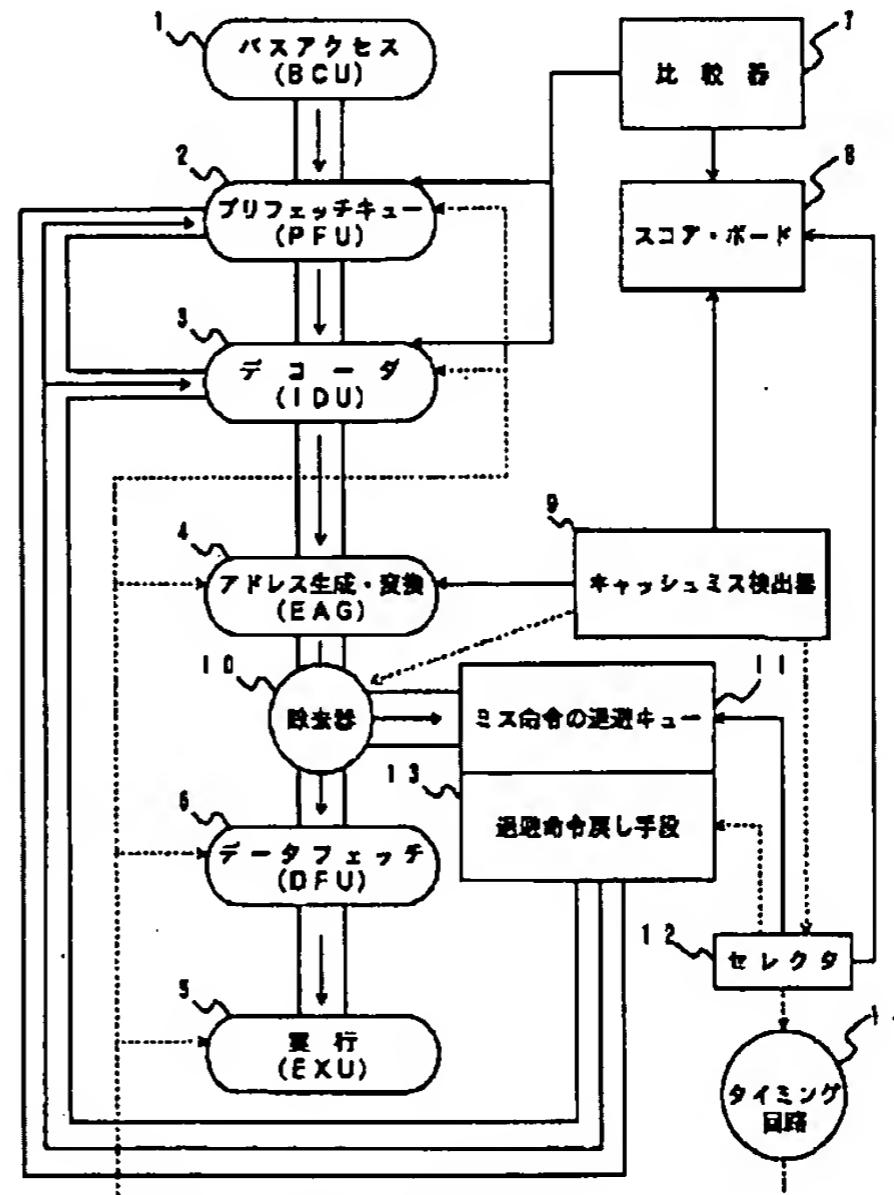
(74)代理人 弁理士 高田 守

(54)【発明の名称】 情報処理装置

(57)【要約】

【目的】 パイプライン命令処理機能をもつ計算機においてスコアボーディング機能を持たせて、命令をデータフロー依存の発生位置までに限り、後方に挿入し直しパイプの中身を消去することなく処理を続けかつキャッシュにデータが読み込まれるまで何サイクルも待つことなく処理することにより、プロセッサにおける処理能力の向上を得る。

【構成】 命令パイプラインが処理されていく過程で、比較器7でプリフェッチ・キュー2及びパイプライン上の複数個の命令のデータフロー依存をスコアボード8に記録し、ある命令セットにおいてキャッシュミス検出器9がキャッシュミスの発生を検出した時その命令セットをいったん退避専用キュー11に格納して次の命令を処理する。その間セレクタ12はスコアボードを調べキャッシュミスが発生した命令とデータフロー依存が発生する命令の位置を検知し、命令の進み方をタイミング回路14により調整しデータフロー依存が発生した命令の前に挿入する。



1

【特許請求の範囲】

【請求項1】 キャッシュメモリと実行する命令を事前にプリフェッチするプリフェッチキューと命令パイプラインを備え、命令実行のパイプライン処理を行なう計算機システムにおいて、
 プリフェッチキューと命令パイプライン中に蓄えられている全ての複数の命令が指定するレジスタオペランド及びメモリオペランドのアドレスを比較する比較手段と、該比較手段の出力から、同一オペランドを指定しているものを検出した場合に、命令の実行順序と同一の順番をオペランドに対して付加し記録する記録手段と、命令パイプライン上の命令のアドレスを生成・変換した段階でキャッシュと比較し該命令にキャッシュミスが発生したか否かを検出す検出手段と、該検出手段の出力を受けてキャッシュミス発生時に該命令をパイプライン上から除去する除去手段と、該除去手段によって命令パイプライン上から除去されたキャッシュミス発生命令を格納しておく格納手段と、上記記録手段によって得られた記録を参照し、上記格納手段によって格納された命令とデータフロー依存が発生している命令のパイプライン上の位置を検出しパイプライン上に戻す位置を選択する選択手段と、キャッシュミス発生時に、命令パイプラインから除去された命令の位置の命令パイプラインの進みと、命令パイプラインに別途格納された命令を挿入する位置のパイプラインの進みを調節するタイミング調整手段と、上記選択手段によって選択された位置に上記タイミング調整手段によって進み方を調整されたパイプライン上に別途格納された命令を戻す戻し手段と、を備えた情報処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、パイプライン処理機能を持つ計算機において、キャッシュミスを発生させた命令をパイプライン上後方へ移動することにより処理を高速化する方法に関するものである。

【0002】

【従来の技術】一般に、32ビット・マイクロプロセッサでは高速化の手段としてパイプライン方式が用いられている。プロセッサの内部はいくつかの機能ブロックのステージから構成され、各ブロックは入力を処理して次のブロックへその結果をわたすようになっている。従来例として、図4にパイプラインの構成図をまた図5にパイプ・ライン処理の概要を示す。この図4と図5は「32ビットマイクロプロセッサ入門」(南宗宏著、CQ出版社)に示されたものである。

【0003】図5において、1はバス・インターフェース(BCU)で、2は命令を溜めておくプリフェッチ・キュー(PFU)で、3は命令をデコードするデコーダ・ユニット(IDU)である。4はアドレスの生成・変

2

換を行なうステージ(EAG)である。5は実行のステージ(EXU)である。

【0004】次にパイプ・ラインの流れの概略を図5で示す。BCU1によるバス・アクセスで読み出された命令が、PFU2のプリフェッチ・キューに溜り、IDU3のデコーダへ渡されてデコードされる。そして、デコーダからアドレスの情報がEAG3へ渡され、アドレスが生成・変換されEXU5で実行される。ここで、プリフェッチ・キュー2には連続する命令がいくつも順番に溜っているものとする。

【0005】図6は命令1から順番にアクセスされる場合のパイプ・ラインの様子を図示したものである。601～608はサイクルはそれぞれ1サイクルに対応する。サイクル601ではバス・アクセス1で命令1が読み出されたところである。次にサイクル602では命令1はデコーダ3に渡され、同時に命令2がバス・アクセス1で読み出されたところである。サイクル603に移ると命令1、命令2は次のステージにそれぞれ移り新しく命令3がフェッチされる。そして、サイクル604、サイクル605の様に順番に各命令は同時にそれぞれのステージで処理されていく。このように1つの命令をいくつかのステージに分割し各ステージの処理を同時にいくつも処理していく機能をパイプ・ライン制御機能という。

【0006】しかし、実際の処理の流れの中では、命令の各ステージの処理時間の間にばらつきがあり、オペランドの読みだし／書き込み、アドレス計算でのメモリ間参照のバス・サイクルの挿入、データ・フェッチでのキャッシュ・ミスの発生などがあって図6のように単純に処理が流れていくわけではない。

【0007】従来技術では、キャッシュ・ミスが発生したときには図5において命令1がサイクル603のところでキャッシュ・ミスを検知するとパイプ・ラインの流れはそこで止まりメモリからキャッシュにデータが読み込まれるのを待ち、メモリからキャッシュに読み込まれた後、順調に処理が行なわれた場合のサイクル604と同じ状態へと移行することにより、パイプ・ラインの内容を消すことを防いできた。

【0008】

【発明が解決しようとする課題】従来技術の手段では、パイプ・ラインの内容を消すことを防ぐ効果はあってもキャッシュ・ミス発生時にパイプ・ラインの流れをデータがメモリからキャッシュ内に読み込まれるまで止めておかなければならぬ。また、メモリからデータをロードするのには、かなり時間がかかり、そのため処理は何サイクルも待たされることになり、その間は無駄に費やされることになる。

【0009】この発明は、上記のような問題点を解消するためになされたもので、キャッシュ・ミス発生時に処理の流れをできるだけ止めることなく行ない、その結果

パイプ・ライン制御機能の処理性能の向上を目的とするものである。

【0010】

【課題を解決するための手段】この発明に係わる情報処理装置は、プリフェッチキューと命令パイプライン中に蓄えられている全ての複数の命令が指定するレジスタオペランドとメモリオペランドのアドレスを比較する比較手段と、該比較手段の出力から、同一オペランドを指定しているものを検出した場合に、命令の実行順序と同一の順番をオペランドに対して付加し記録する記録手段と、命令パイプライン上の命令のアドレスを生成・変換した段階でキャッシュと比較し該命令にキャッシュミスが発生したか否かを検出する検出手段と、該キャッシュミス検出手段の出力を受けてキャッシュミス発生時に該命令をパイプライン上から除去する除去手段と、該除去手段によって命令パイプライン上から除去されたキャッシュミス発生命令を格納しておく格納手段と、上記記録手段によって得られた記録を参照し、上記格納手段によって格納された命令とデータフロー依存が発生している命令のパイプライン上の位置を検出しパイプライン上に戻す位置を選択する選択手段と、キャッシュミス発生時に、命令パイプラインから除去された命令の位置の命令パイプラインの進みと、命令パイプラインに別途格納された命令を挿入する位置のパイプラインの進みを調節するタイミング調整手段と、上記選択手段によって選択された位置に上記タイミング調整手段によって進み方を調整されたパイプライン上に別途格納された命令に戻す手段と、を設けたものである。

【0011】

【作用】この発明においては、キャッシュミス検出手段によってキャッシュミスを検出すると出力信号によりその命令をパイプライン上から除去手段により除去し、格納手段に格納し、同時に検出手手段の出力信号を受けて選択手段は記録手段を参照し、格納手段に格納された命令に戻す位置をパイプライン上で決定し、タイミング調整手段によりパイプラインの進み具合を調整し、格納されていた命令を戻し手段によりパイプライン上に戻す。

【0012】

【実施例】以下、この発明の一実施例を図面について説明する。また、図1においては図5の従来例と同一または相当部分には同一符号を用いている。本実施例では命令パイプラインの先頭の命令(命令1)でキャッシュミスが発生し、命令1は3つ後方にある命令4とデータフロー依存が発生しているものとする。

【0013】図1中、1はバス・インターフェース(BCU)、2はプリフェッチ・キュー(PFU)、3はデコーダ(IDU)、4はアドレス生成・変換(EAG)、5は実行(EXU)の各ステージでそれぞれ図5に対応しており、処理の流れも前述の通りである。また6はデータフェッチのステージで、EAG4で生成・変

換されたアドレスをもとにデータをフェッチするステージである。7はバス・アクセスによって読み出される全ての命令のレジスタ・オペランド及びメモリアドレスを比較してそれまでに読み出されたどの命令とデータフロー依存が発生しているかを調べる比較器である。8は比較器7で調べた結果を保持するスコアボードである。9はEAG4で生成・変換されたアドレスを調べキャッシュミスが発生したかどうかを調べる。10はEAG4でキャッシュミスが発生した場合、キャッシュミス検出器10からの出力信号を受けて、ミス命令を格納手段に送りパイプライン上から命令を除去する除去器である。11は除去器10によってパイプライン上から除去された命令を格納しておくためのミス命令退避キューである。12はキャッシュミス検出器からの出力信号を受け、スコアボード8を調べミス命令とデータフロー依存が発生している命令のパイプライン上の位置を検出するセレクタである。13はセレクタ12の出力信号を受けてミス命令の退避キュー11に格納されたミス命令をパイプライン上の指定された位置に戻す手段である。14はセレクタ12の出力信号を受けて格納されている命令をパイプライン上に戻す位置から前にある命令を1つ進め後ろの命令を止め該ミス命令に戻す位置を与えるタイミング調整回路である。

【0014】図2は図1の実施例に使用されるスコアボードの構造図である。図2ではプリフェッチ・キュー2に5つ命令が蓄えられているものとするが、当然その数は可能な限り増設できる。図2では、201にはEAG4の命令と後方4命令との間のデータフロー依存の発生を表すデータで、201行211列は1つ前、201行302列は2つ前、201行216列は6つ前の命令との間のデータフロー依存の発生を表すデータが比較器の出力により記録されている。図2の実施例ではデータは順番に0、0、1、0、0、1でEAG4にある命令が3つ前のPFU2の先頭から2番目にある命令及びPFU2の最後尾にある命令とデータフロー依存が発生していることを示す。すなわち、EAG4でキャッシュミスを発生した命令を命令1とすれば命令1は命令4及び命令7とデータフロー依存が発生していることを示す。データの更新はバスアクセスにより命令が読み込まれる度に行なわれ、図2でいうと下方向へシフトし新しく読み込まれた命令との間のデータフロー依存の状態を表すフラグが立てられる。この時図2中216列の所が全ての行にわたって更新される。またデータフロー依存の発生を記録しうる範囲は、先頭がEAG4にある命令で、最後尾はPFU2内の最後尾が限界でそれ以上のデータフロー依存の発生は記録されない。

【0015】次に、上記のように構成されたパイプ・ライン、及びスコア・ボードにおいてキャッシュミス発生時における処理の動作について、図3のフローチャート50を参照しながら説明する。

5

【0016】キャッシュミスが発生したらステップ301でキャッシュミス検出器はスコアボードを参照し、もしその命令とデータフロー依存がn+1番目の命令との間に発生しているならば何もしない。もしそれ以外ならば、ステップ302で除去器に信号を送り命令の流れを切替えn番目の命令をミス命令退避キュー10に送る。

【0017】ステップ303でセレクタ11はスコアボード8を参照し、除去した命令より後方のどの命令にミス命令退避キューとデータフロー依存が発生しているかを調べる。もしあれば、ステップ304でキャッシュミス検出器9の出力信号を受けてセレクタが挿入位置を決定して、セレクタの出力信号を受けてタイミング回路14はその位置から前方の命令を前に一つ進め、それより後方の命令を止めて空いた位置にパイプライン上に格納した命令を戻す手段13によって挿入する。もし、データフロー依存が発生する命令がスコアボード上から検出されなければ、パイプライン上の全ての命令を一つ前に進め、バスアクセスによる命令の読み込みを止め、PFU2の最後尾に格納された命令を戻す。

【0018】次にキャッシュ・ミスが発生した場合の命令の移動する例を図4を使って説明する。図4の401～407はそれぞれ1サイクルに相当する。図4は命令1でキャッシュミスが発生し命令1とデータフロー依存が発生する命令は命令4であるとした場合の図である。

【0019】サイクル403で命令1にキャッシュミスが発生するとキャッシュミス検出器11により、スコアボードを参照し命令2との間にデータフロー依存が発生していないかどうかを調べる。本例では命令1と命令2の間にキャッシュミスは発生していないものとし、サイクル404で命令1はキャッシュミス検出器の出力信号を受け10の除去器によりパイプライン上のミス発生命令1はパイプライン上から除去され、ミス命令退避キュー11に格納される。他のパイプライン上の命令はそれぞれ1つ前に進む。セレクタ12は上記キャッシュミス検出器9の出力信号を受けスコアボードを参照しPFU2及びIDU3上からミス発生命令1とデータフロー依存の発生する命令4を検出し格納された命令の挿入位置を決定する。挿入位置が決定されると次のサイクル405に移行するところで、タイミング回路14はセレクタ13からの出力信号を受け、命令4より前の命令を1つ進め、命令5より後方の命令を止め、ミス発生退避キュー

に格納された命令1の挿入位置をつくる。同じくセレクタ12からの出力信号を受けたパイプライン上に戻す手段13はミス命令退避キューに格納され後続の命令1をサイクル405で命令4の前に挿入する。

【0020】図4からわかるように、キャッシュミスが発生しても1サイクルしか命令に遅滞が生じない。

【0021】

【発明の効果】以上のようにこの発明によれば、ある命令がキャッシュミスを発生すると、それを検出した検出手段の出力信号により、選択手段が比較手段から得られた記録手段を参照しデータフロー依存の発生する命令の位置を見つけ、タイミング調整手段でパイプライン上の命令の進行速度を調整することにより、次のサイクルで該命令をデータフロー依存が発生する位置まで後方に移動し、キャッシュ内にデータが読み込まれるまで待つことなく、1サイクル処理を無駄にするだけで命令のパイプラインの処理を続けることができ、それによってパイプライン機能を持つ計算機の処理能力の向上をはかることが可能となる。

【図面の簡単な説明】

【図1】この発明の実施例によるパイプラインの構成のブロック図である。

【図2】この発明におけるスコアボードの構造図である。

【図3】この発明における動作のフローチャート図である。

【図4】この発明におけるキャッシュミスが発生した場合のパイプライン処理の概要図である。

【図5】従来技術によるパイプラインの構成図である。

【図6】従来技術によるパイプライン処理の概要図である。

【符号の説明】

7 比較手段

8 記録手段

9 キャッシュミス検出手段

10 命令除去手段

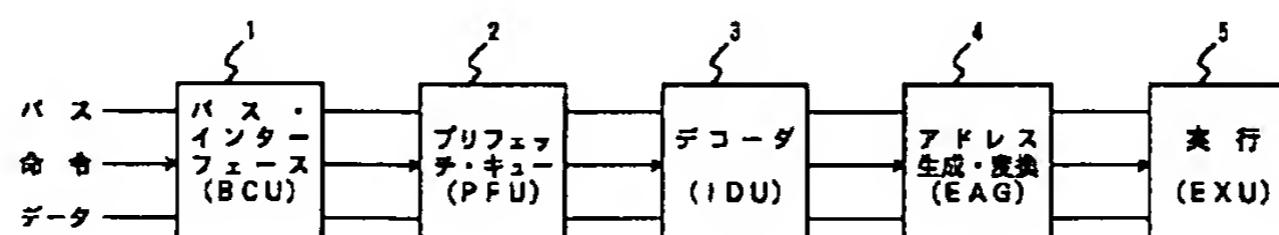
11 キャッシュ命令格納手段

12 選択手段

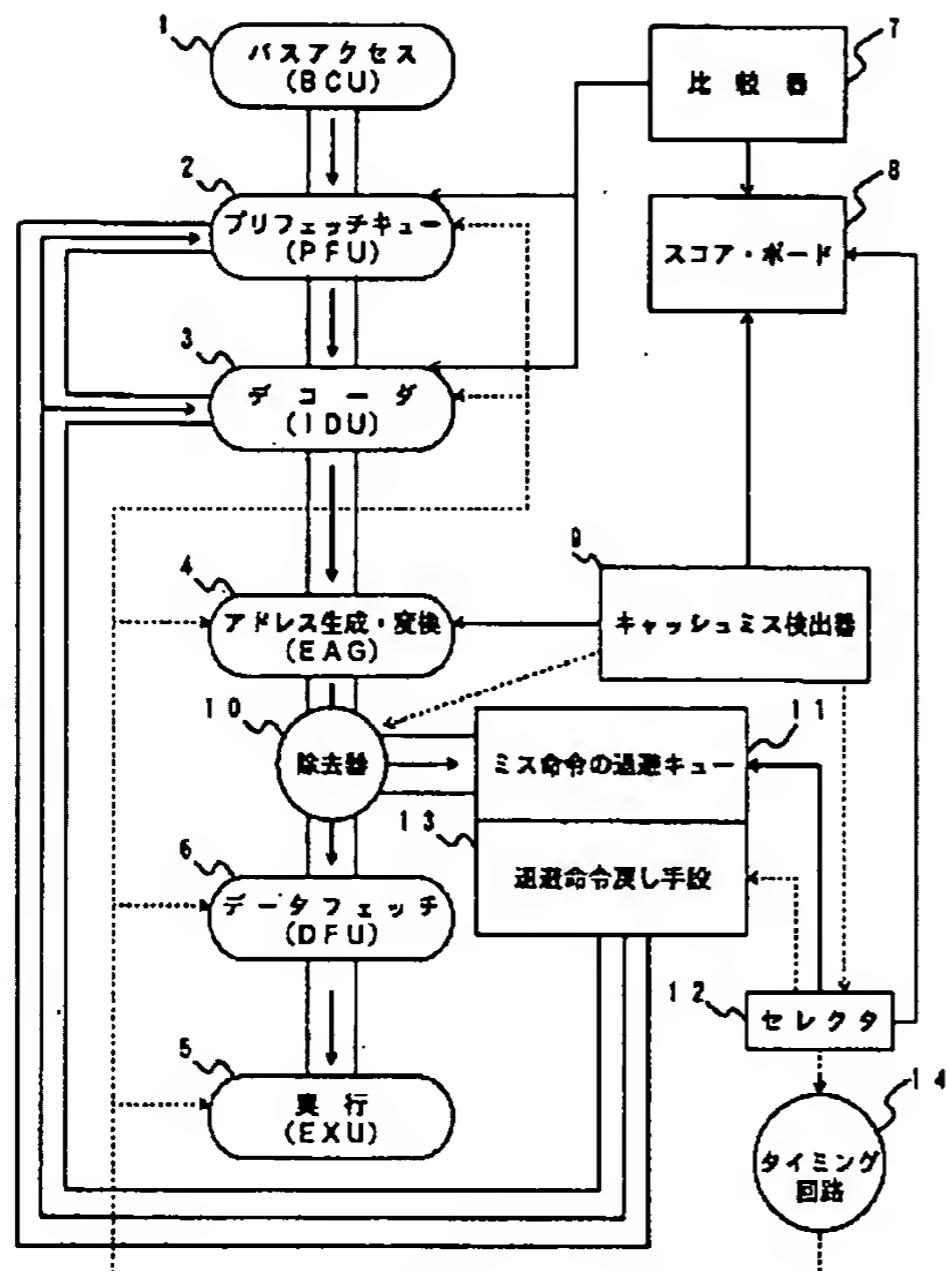
13 退避命令戻し手段

40 14 タイミング調整手段

【図5】



【図1】

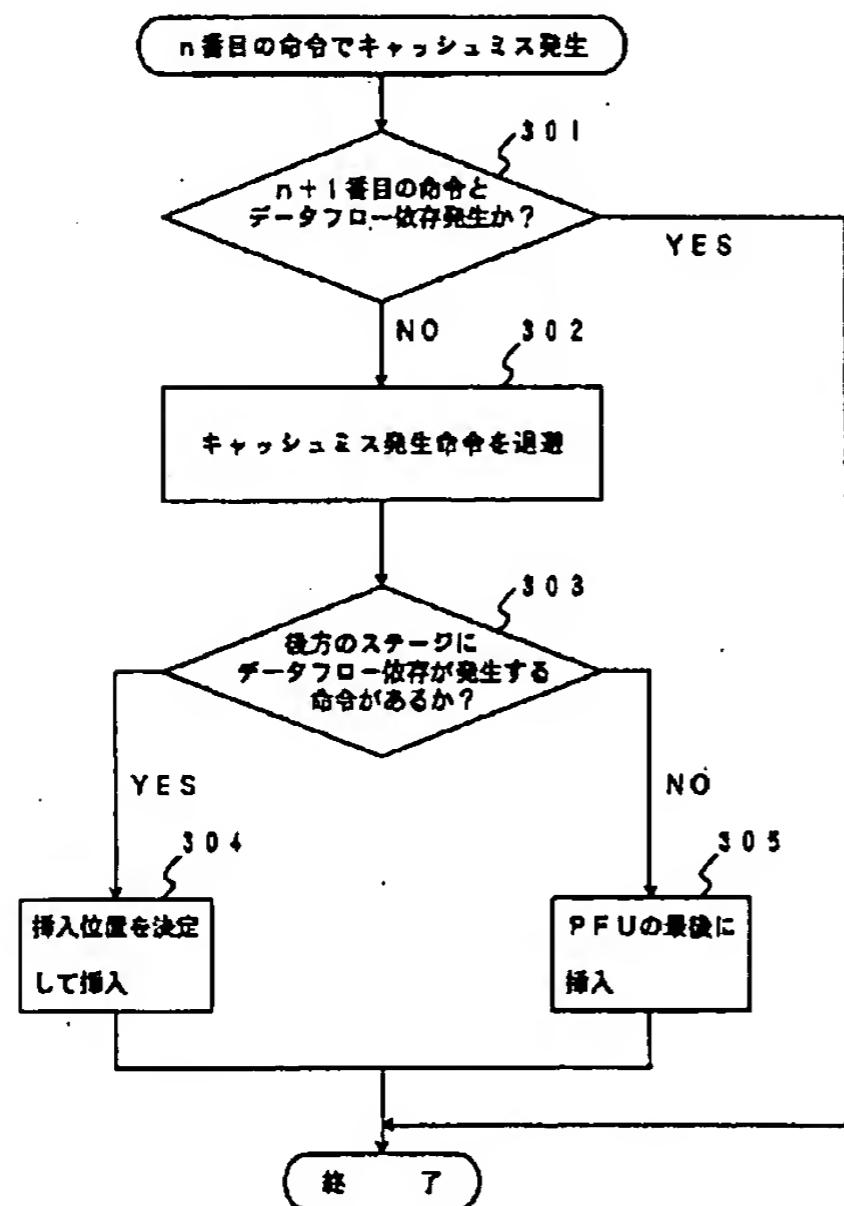


【図2】

207	0	0	0	0	0	0
206	0	0	0	0	0	0
205	0	0	0	0	0	0
204	1	0	0	0	0	0
203	0	0	1	0	0	0
202	0	0	0	1	1	0
201	0	0	1	0	0	1
	211	212	213	214	215	216

【図2】時刻順序図

【図3】



【図4】

	401	402	403	404	405	406	407
BCU	命令1	命令2	命令3	命令4	命令4	命令5	命令6
I DU		命令1	命令2	命令3	命令1	命令4	命令5
E AG			命令1	命令2	命令3	命令1	命令4
D FU					命令2	命令3	命令1
EXU						命令2	命令3

【図6】

→ 時間

	601	602	603	604	605	606	607	608
バス・ アクセス	命令1	命令2	命令3	命令4	命令5	命令6	命令7	命令8
デコード		命令1	命令2	命令3	命令4	命令5	命令6	命令7
アドレス の生成 ・変換			命令1	命令2	命令3	命令4	命令5	命令6
実 行				命令1	命令2	命令3	命令4	命令5